CLAIMS

What is claimed is:

1. A data modem apparatus comprising a first processor for receiving and processing a serial stream of analog data over time wherein the first processor is programmed to convert the analog data to digital data and configures the digital data into data packets, the first processor being adapted to be coupled to a bus interface so that the data packets may be passed from the first processor to a second processor via the bus interface for further digital processing of the data packets into byte-ordered information.

2. The data modem apparatus of claim 1 wherein the first processor is a digital signal processor (DSP) designed to convert data between the time domain and the frequency domain to create the data packets from the serial stream of data.

3. The data modem apparatus of claim 1 wherein the second processor is a host processor for a computer wherein the host processor is coupled to the first processor by a data bus comprising address lines, data lines, and at least one control line.

4. The data modem apparatus of claim 1 wherein the data modem apparatus processes two or more of G.lite data streams, V.90 data streams, and ADSL data streams.

5. The data modem apparatus of claim 1 wherein the data modem apparatus processes all three of G.lite data streams, V.90 data streams, and ADSL data streams.

6. The data modem apparatus of claim 1 wherein the second processor is coupled to assemble and/or dissasemble ATM data packets which are then configured for transmission over as DMT symbols.

7. The data modem apparatus of claim 1 wherein the data modem apparatus has two modes of operation, a first mode of operation being where the second processor dedicates X MIPS to processing the data packets and a second mode of operation being where the second processor dedicates Y MIPS to processing the data packets where Y is less than X.

8. The data modem apparatus of claim 1 wherein the data modem apparatus has three modes of operation, a first mode of operation being where the second processor dedicates X MIPS to processing the data packets, a second mode of operation being where the second processor dedicates Y MIPS to processing the data packets where Y is less than X but nonzero, and a third mode of operation where the processor devotes no MIPS to processing the data packets.

9. The data modem apparatus of claim 1 wherein a buffer is coupled between the first processor and the second processor wherein a plurality of data packets are queued into the buffer before the second processor begins to process the plurality of data packets.

10. The data modem apparatus of claim 1 wherein a buffer is coupled between the first processor and the second processor wherein a plurality of data packets are queued into the buffer before the first processor begins to process the plurality of data packets.

11. The data modem apparatus of claim 10 wherein the first processor transmits one or more redundant symbol if the buffer contains no new valid data to transmit.

12. The data modem apparatus of claim 11 wherein the first processor monitors a number N of consecutive redundant symbols and correctively prevents the transmission of N newly provided symbols to offset the previously-transmitted N consecutive redundant symbols.

13. A modem apparatus comprising:

an analog interface for interfacing to a communication line;

a digital signal processor coupled to the analog interface, the digital signal processor being coupled performs frequency domain equalization (FEQ) operations, time domain equalization (TEQ) operations, fast fourier transform (FFT) operations, inverse fast fourier transform (iFFT) operations, and encoding/decoding operations on a serial stream of data provided through the analog interface, the digital signal processor having an output;

a data bus coupled to the output of the digital signal processor; and

a host central processing unit (CPU) coupled to the data bus for receiving packets of data from the digital signal processor, the host CPU performing error correction operations on the data within the packets of data.

14. The modem apparatus of claim 13 wherein the host CPU also performs interleaving and/or deinterleaving operations to intermingle or separate two or more different data symbols.

15. The modem apparatus of claim 13 wherein the host CPU performs Reed Solomon encoding and/or decoding on the data within the data packets.

16. The modem apparatus of claim 15 wherein the host CPU performs cyclic redundancy check (CRC) operations on the data within the data packets.

17. The modem apparatus of claim 13 wherein the host CPU performs tone ordering and/or deordering on the data within the data packets.

18. The modem apparatus of claim 13 wherein the host CPU configures and processes ATM data for transmission within DMT data packets.

19. The modem apparatus of claim 13 wherein the digital signal processor performs QAM encoding and/or decoding operations.

20. A method comprising the steps of:

receiving a serial stream of data via a first processor;

performing time domain to frequency domain operations on serial stream of data to create digital values via the first processor;

performing frequency domain equalization on the digital values via the first processor;

decoding the digital values to a binary stream that is packaged into byte/word boundaries of the first processor;

providing the binary stream to a second processor;

deinterleaving data within the binary sream via the second processor; and

error code correcting and/or CRC processing the binary stream via the second processor.

21. The method of claim 20 wherein the first processor ATM processes the binary stream to extract data from ATM cells.

22. The method of claim 21 wherein the first processor processes the binary stream to extract data from TCP/IP packets.

23. The method of claim 20 wherein both the first and second processors control buffers that are used to buffer the binary stream between the first processor and the second processor.

24. The method of claim 20 wherein steps performed by the first processor can be dynamically switched over to the second processor.

25. The method of claim 20 wherein steps performed by the second processor can be dynamically switched over to the first processor.

26. The method of claim 20 wherein processing in the second processor is involved only after a certain amount of the binary stream is provided from the second processor to the first processor.

27. A method comprising the steps of:

performing a first plurality of digital operations using a first processor in order to process an incoming stream of digital information into a first digital format;

performing a second plurality of digital operations using a second processor in order to process the first digital format into a second digital format; and

reassigning the first and second plurality of digital operations between the first processor and the second processor in order to

dynamically alter the performance load of one of either the first or second processor over time.

28. A method for communicating DSL data symbols, the method comprising the steps of:

receiving one or more DSL data symbols over time and buffering the DSL data symbols in a buffer;

retrieving DSL data symbols from the buffer and transmitting the DSL data symbols one after another along a communication line;

retransmitting, when the buffer contains no DSL data symbols, a redundant DSL symbol along the communication line; and

correcting, when the buffer eventually is written to contain DSL data symbols, a state of DSL transmission to compensate for the previous communcation of redudant symbols.

29. The method of claim 28 wherein the DSL symbols contain TCP/IP packets that are assembled into ATM data cells.